

## P2.17 CLOUD CLASSIFICATION IN POLAR AND DESERT REGIONS AND SMOKE CLASSIFICATION FROM BIOMASS BURNING USING A HIERARCHICAL NEURAL NETWORK

June Alexander\*, Edward Corwin, David Lloyd, Antonette Logar and Ronald Welch

South Dakota School of Mines and Technology  
Rapid City, South Dakota

### 1. INTRODUCTION

This research, funded by NASA grant NAS 1-19077 and by NASA Space Grant NGT 40046, focuses on a new neural network scene classification technique. The task is to identify scene elements in Advanced Very High Resolution Radiometry (AVHRR) data from three scene types: polar, desert and smoke from biomass burning in South America (smoke). The ultimate goal of this research is to design and implement a computer system which will identify the clouds present on a whole-Earth satellite view as a means of tracking global climate changes. Previous research has reported results for rule-based systems (Tovinkere *et al* 1992, 1993) for standard back propagation (Watters *et al.* 1993) and for a hierarchical approach (Corwin *et al* 1994) for polar data. This research uses a hierarchical neural network with don't care conditions and applies this technique to complex scenes.

A hierarchical neural network consists of a switching network and a collection of leaf networks. The idea of the hierarchical neural network is that it is a simpler task to classify a certain pattern from a subset of patterns than it is to classify a pattern from the entire set. Therefore, the first task is to cluster the classes into groups. The switching, or decision network, performs an initial classification by selecting a leaf network. The leaf networks contain a reduced set of similar classes, and it is in the various leaf networks that the actual classification takes place. The grouping of classes in the various leaf networks is determined by applying an iterative clustering algorithm. Several clustering algorithms were investigated, but due to the size of the data sets, the exhaustive search algorithms were eliminated. A heuristic approach using a confu-

sion matrix from a lightly trained neural network provided the basis for the clustering algorithm. Once the clusters have been identified, the hierarchical network can be trained. The approach of using don't care nodes results from the difficulty in generating extremely complex surfaces in order to separate one class from all of the others. This approach finds pairwise separating surfaces and forms the more complex separating surface from combinations of simpler surfaces.

This technique both reduces training time and improves accuracy over the previously reported results. Accuracies of 97.47%, 95.70%, and 99.05% were achieved for the polar, desert and smoke data sets.

### 2. THE DATA

The raw satellite data for each of the three scene types, polar, smoke and desert, was converted into thirty-element feature vectors by combining information from five satellite channels. The resolution for each pixel of AVHRR data is 1.1 km. at nadir. The first five elements consist of the original readings from channel 1 (0.56 - 0.68  $\mu\text{m}$ ), channel 2 (0.72 - 1.1  $\mu\text{m}$ ), channel 3 (3.55 - 3.93  $\mu\text{m}$ ), channel 4 (10.3 - 11.3  $\mu\text{m}$ ) and channel 6 (11.5 - 12.5  $\mu\text{m}$ ). Elements 6 through 15 are channel ratios and elements 16 through 25 are channel differences. The remaining five elements are computed by subtracting the average of the eight neighboring pixels (for each of the five channels) from the center pixel.

The task is to classify the pixels in the image as one of 22 previously determined classes. These classes are listed in Table 1. Obviously, not all classes can occur in every scene. For polar scenes, pixels were identified as belonging to classes 1, 2, 3, 4, 5, 6, 8, 12, or 21. For smoke scenes the possible classes are 1, 3,

\* *Contributing Author's Address:* Institute of Atmospheric Sciences, South Dakota School of Mines and Technology, 501 East Saint Joseph Street, Rapid City, SD 57701-3995.

class 1	water
class 2	snow and ice
class 3	ice cloud
class 4	land
class 5	thin water cloud
class 6	stratus water cloud
class 7	cumulus over water
class 8	textured snow/ice or broken sea ice
class 9	snow covered mountain
class 10	multilayered cloud
class 11	smoke over land
class 12	water cloud over land
class 13	cumulus over land
class 14	desert
class 15	water cloud over desert
class 16	thin water cloud over land
class 17	sunlint water, reflectivity < 11
class 18	sunlint super, reflectivity >= 11
class 19	dust over ocean
class 20	dust over land
class 21	slush
class 22	silty water

Table 1 : The Classes

4, 6, 11, 12, 17, and 22. For desert data the classes are 1, 3, 4, 5, 6, 12, 14, 15, 17, 18, 19, and 20.

The data was divided into training and testing sets to allow for the use of a supervised learning neural network training algorithm. Approximately 10% of the data is selected at random for training and the remainder is used for testing. Specifically, for the polar data 3,326 vectors were used for training and 28,693 for testing, for the desert data 2,344 were selected for training and 17,976 for testing and for the smoke data 3,249 were used for training and 29,036 for testing. The deviation from precisely 10% is the result of requiring a minimum of 100 vectors from each class. The resulting training vector files were shuffled to ensure that the elements are presented to the network in a random order. This can greatly reduce training time and improve training accuracy.

The normalization technique used has a large impact on training. The best results were obtained by using :

$$x_{new} = (x_{original} - x_{minimum})/x_{range}$$

This technique requires knowledge of both the minimum value for each component in the vector and the range of values for that component. The minimum and range values were computed from the data for a

given scene type. Thus, different constants are required for polar, smoke and desert. It is generally accepted that the testing data should not be used in any way in the training process. However, in this situation, the testing data is more of a validation set. It is a set of data kept for testing the training of the network but it is not the ultimate testing data. Scenes with several million pixels are the true test set. For this reason, we felt it was justified to use the entire data set in calculating the normalization constants.

Note also that it is preferable to normalize the entire data set with the same normalization parameters. However, due to the large differences in the sizes of elements, column-wise normalization was necessary. In some sense, this does lose the information contained in the relative sizes of the elements but the loss of information due to inaccuracies caused by working with very small numbers greatly outweighed this concern.

### 3. HIERARCHICAL NEURAL NETWORKS WITH DON'T CARE CONDITIONS

The hierarchical approach is essentially a divide-and-conquer strategy. The classes are grouped into clusters and a two stage classification process is used. The first stage identifies which cluster a vector

belongs to, and the second stage determines a class assignment within the cluster. The stage which selects a cluster is called the switching network. The switching net used in this research is a standard back propagation network with 30 inputs, 20 hidden nodes and a number of outputs corresponding to the number of clusters for a given data set. The learning rate was kept constant at 0.1 and the momentum rate at 0.5 for all hierarchical experiments.

Several clustering techniques were applied to a subset of the data but all proved unacceptably computationally intensive when used on the full data set. A heuristic clustering approach was developed which produced good clusters and required a fraction of the time needed for the other algorithms. The fundamental difference is that the clustering is done on the classes rather than on individual vectors. The technique is to train a back propagation network for 50 iterations and generate a confusion matrix. The process of assigning clusters is then done as follows :

1. Find all classes which have no "conflicts" and put them into a holding area. A class is considered to have no conflicts if no entry in the confusion matrix is greater than 10% for that class. This indicates that the class is easily separated from the other classes and can go into any cluster.
2. For each of the remaining classes, identify the entries in the confusion matrix which are greater than 10% and place all of these classes into the same cluster. Repeat this process until all classes are assigned to a cluster.
3. Balance the clusters by putting in classes from the holding area, or, if none are needed, the holding area becomes its own cluster.

The second step in the network classification is to choose a class from among those in the cluster selected. The network used for this task, called the leaf network, is a back propagation network modified to incorporate don't care conditions (Logar *et al* 1994). Although the number of input and hidden nodes was kept the same as in the switching network, the number of output nodes is dependent upon the number of classes in a cluster. If  $n$  is the number of classes in a cluster, the size of the output pattern for a don't care network is  $n*(n-1)/2$ , the number of pairwise combinations of the classes. The don't care network then builds the complete separating surface from a combination of these pairwise separators. Since the pairwise separating

surfaces are easier to construct, training time is reduced.

One of the interesting features of a don't care algorithm is the ability to return a value which indicates that the network output did not match any of the valid output patterns. For some applications, it may be desirable to simply indicate that the classification failed. For this application, failure to classify a pixel was deemed unacceptable. Thus, the algorithm was modified to select the "closest" classification. One technique implemented was to assume that an unclassified pixel was the same class as the closest pixel previously classified. This produced streaks in the image and proved unacceptable. A better approach was to compute the Euclidean distance between the network output and all valid output patterns and select the class corresponding to the valid output pattern with the smallest distance.

The hierarchical network, as well as the back propagation network used for baseline data, incorporated weight projections (Logar *et al* 1992). Weight projections fit a weighted least squares quadratic curve to the trajectory of each weight in the network and project future weight values from the trajectory. The jumps save approximately half of the training time. If a jump is not advantageous, as determined by an increase in the training error, it is undone and training proceeds from that point as it would have without the jump. The weighting refers to giving points later in the trajectory more weight in determining the fit of the curve. This addition improved the number of jumps that were kept during a training run and allowed for larger jumps to be made.

#### 4. RESULTS

Experiments were conducted using both back propagation networks and hierarchical networks with don't care conditions. The results, summarized in Tables 2 and 3, show that considerable gains in accuracy were achieved using the hierarchical scheme. The increase in accuracy can be attributed to two features of this topology. First, the number of classes in each cluster and in the switching network is significantly smaller than for a single monolithic back propagation network, thereby simplifying the classification task and improving performance. In addition, since each network is small, training time is also reduced. In fact, all of the leaf networks and the switching network can be trained independently, and thus in parallel, for significant reductions in training time.

	Training	Testing	Overall	Collapsed
Polar	92.26	89.26	89.58	92.46
Desert	81.96	83.35	83.21	90.01
Smoke	97.72	96.71	96.81	98.97

Table 2 : Back Propagation Results

	Training	Testing	Overall	Collapsed
Polar	92.60	91.56	94.07	97.47
Desert	93.22	90.24	92.08	95.70
Smoke	98.69	97.58	98.19	99.05

Table 3 : Hierarchical Network  
with Don't Care Conditions Results

The second source of improvement is from the don't care nodes. The separating surface is a composition of pairwise separating surfaces which can be found with greater accuracy than can a single separating surface which must distinguish one class from all others. Again, that simplicity leads to a reduction in training time. A disadvantage to the don't care technique, however, is the increased storage requirements. As stated previously, the number of output nodes required to represent  $n$  classes is  $n * (n-1)/2$ . This increases the number of weights into the hidden layer from  $n*m$ , where  $m$  is the number of hidden nodes, to  $n*m*(n-1)/2$ . This disadvantage is mitigated by the small leaf and switching network sizes.

The tables present four values for each of the three scene types. The training number indicates the accuracy achieved by the network at the cessation of training. The testing data is achieved by presenting the previously unseen data to the trained network. The overall number is the result of presenting the entire data set to the trained network. Since the major focus of the project is to distinguish between cloud and non-cloud scene elements, it was determined that several similar classes could be combined into a single class. The collapsed number is the result of treating thin water cloud, stratus water cloud, water cloud over land and water cloud over desert as a single class and by treating snow /ice and textured snow and ice and broken sea ice as a single class. The networks were trained assuming all classes to be unique, but the collapsed network accuracy was computed by treating a misclassification as correct if it was misclassified within its group. For example, a broken sea ice vector classified as snow/ice would be considered correct since these classes are collapsed. Separating the classes for training reduced training time and improved classification accuracy.

The standard back propagation networks all contained 30 input nodes corresponding to the size of the input vector as described above. Each network had a single hidden layer with 20 nodes. Not all classes can occur in every scene, thus, the number of output nodes depended upon the type of scene being analyzed. The number of output nodes for polar, desert and smoke are 9, 12, and 8 respectively. The learning rate was kept constant at 0.1 and the momentum rate at 0.5 for all back propagation experiments.

In all scenes, the back propagation network mistakenly classifies land, sunglint and water as cloud. The result is to exaggerate the amount of cloudiness in each scene. The hierarchical network gives a much more accurate representation of the scene. Both networks have difficulty in transition areas, especially the transitions between cloud and water. In these areas, land or sunglint appears at the transition in the smoke and desert scenes. In the polar scenes, slush and textured snow and ice appear in the transition areas. Sunglint was difficult to identify in transition areas as well. In the transition between sunglint and water, pixels were mistakenly classified as cloud or dust over land. However, the number of sunglint samples was small and the problem may be corrected when additional samples become available.

## 5. CONCLUSION

The hierarchical network approach described here is an effective tool for complicated scene classification. Future research in this area will center on feature selection and clustering, since these areas provided the greatest challenge. In addition, a network is being built which will use a variable number of inputs at each stage to refine the classification process. Using differ-

ent vector elements for the switching network and the leaf networks may increase classification accuracy.

## 6. ACKNOWLEDGMENTS

Support for this work was funded by NASA grant NAS 1-19077 and by NASA Space Grant NGT-40046

## 7. REFERENCES

Corwin, Edward M., Sterling Greni, Antonette Logar, Karen Whitehead and Ronald Welch, "A Multi-Stage Neural Network Classifier", *Proceedings of the World Congress on Neural Networks*, June 1994.

Logar, Corwin, Oldham, "An Efficient Gradient Descent Learning Scheme Using Weight Trajectory Projections", *Proceedings of the Oklahoma Artificial Intelligence Conference*, Nov. 1992.

Logar, Corwin, Watters, Weger and Welch, "A Don't Care Back Propagation Algorithm Applied to Satellite Image Recognition", *Proceedings of National SAC/ACM Conference*, March 1994.

Tovinkere, V., "Fuzzy Logic Expert Systems for Classification of Polar Regions", *M.S. Thesis, Department of Mathematics and Computer Science, South Dakota School of Mines and Technology*, 1992.

Tovinkere, V., Manuel Penaloza, Antonette Logar, Jonathan Lee, Ronald Weger, Todd Berendes, and Ronald Welch, "An Intercomparison of Artificial Approaches for Polar Scene Identification", *Journal of Geophysical Research*, Vol. 98, March 1993.

Watters, S.E., Logar, A. M., Corwin, E. M., "A Comparison of Neural Network Algorithms for Polar Scene Classification", *Intelligent Engineering Systems Through Artificial Neural Networks*, vol. 3, Nov 1993.

